Theoretical Computer Science 74 (1990) 163-181
North-Holland

AUTOMATA REPRESENTED BY PRODUCTS OF SOLITON AUTOMATA *

F. GÉCSEG

József Attila Tudományegyetem Bolyai Intézete, Aradi vértanúk tere 1, H-6720 Szeged, Hungary

H. JÜRGENSEN

Department of Computer Science, The University of Western Ontario, London, Ontario, Canada N6A 5B7

Communicated by A. Salomaa
Received November 1987
Revised August 1988

Abstract. Soliton valves have been proposed as molecular switching elements. A mathematical model of the logic aspects of soliton switching, called soliton automaton, was introduced by Dassow and Jürgensen. They analysed the simulation power of strongly deterministic soliton automata with respect to transition semigroups. Here, a more detailed analysis in terms of homomorphic and isomorphic representations with respect to various automaton products is given.

1. Introduction

There are various speculations about the direction of the development of future computer architecture. In [3, 4] a mathematical model was provided, and several mathematical results were proved which could be used to determine the computational power of one type of a proposed switching device, the "soliton valve" (see [1]).

Research in bioelectronics has proposed several chemical structures as basic building blocks for future computers. For a survey see [1] and the proceedings volume [2]. Among these, "soliton valves" seem to be very interesting candidates. Their switching behaviour is based on the effects of a soliton wave travelling along a molecule chain. The main example discussed in the literature works with polyacetylene chains as shown in Fig. 1. Ignoring the physico-chemical details, the effect of

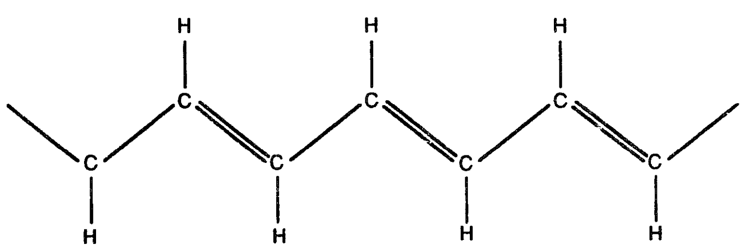

Fig. 1. (CH)-chain.

* This work was supported by Natural Science and Engineering Research Council of Canada, Grant A0243.

0304-3975/90/$03.50

a soliton wave propagating along such a chain is to exchange all single and double bonds. In terms of switching logic this amounts to the action of a flip-flop.

In this paper we are only interested in the logic aspects of "soliton valves". For the physico-chemical background see [6]. We adopt the definitions of [3] and [4] of "soliton graphs" and "soliton automata" based on soliton graphs as a mathematical model of "soliton valves". "Strongly deterministic" soliton automata form a class of particular interest.

It seems to be a natural and quite important question to determine the computational power of soliton automata. One approach is to study the transition monoids of soliton automata [3-5]. In [3] and [4] it is shown that the transition monoids of strongly deterministic soliton automata are direct products of primitive groups generated by involutions. Furthermore, every symmetric group is the transition monoid of some strongly deterministic soliton automaton. On the other hand, not every transition monoid of a strongly deterministic soliton automaton—considered as a permutation group rather than an abstract group—is a symmetric group. However, it is an open question, whether the transion monoid is isomorphic or not isomorphic with a symmetric group in such cases. Thus, as is mentioned in [3] and [4], if the transition monoid is used as a yardstick for computational power, strongly deterministic soliton automata are insufficient to simulate all finite automata. As all "small" strongly deterministic soliton automata give rise to the respective symmetric groups only, the simulation of arbitrary "small" group automata with respect to transition monoids may be achievable only at forbidding costs. The smallest transition monoid of a strongly deterministic soliton automaton, which is not a symmetric group, has at least 9! and no more than 12!/2 elements! The implications of these complexity considerations are not yet fully understood.

In this paper we follow a different approach to analyse the computational power of soliton automata. Instead of simulation with respect to transition monoids we study actual automaton simulation. A typical question addressed is the following: given a class $\mathscr{C}$ of strongly deterministic soliton automata, which automata can be represented homomorphically or isomorphically by products of automata in $\mathscr{C}$.

The products to be considered in the subsequent chapters are as follows: α_0-products, that is, products without feed-back; α_i-products for $i \geq 1$, that is, products with feed-back loops whose length is bounded by i; general products, that is, products with arbitrary feed-back loops [10]. In Section 4, we show that all commutative permutation automata can be represented homomorphically by α_0-products of strongly deterministic soliton automata. Clearly, however, not every automaton can be represented in this way. On the other hand, α_1-products already suffice to represent all automata homomorphically; this is shown in Section 5. Then, in Section 6 we prove a characterization of homomorphically complete classes of strongly deterministic soliton automata with respect to α_i-products, $i \geq 2$, and general products. Finally, in Section 7, we show that no class of strongly deterministic soliton automata is isomorphically complete with respect to any α_i-product. A few conclusions are summarized in Section 8.

Of the remaining sections, Section 2 reviews basic notions and notation; in particular, the required definitions from automaton theory are provided. The necessary definitions concerning soliton automata are given in Section 3. For additional details, the references [10, 11, 1, 3] should be consulted.

2. Basic notions

In this section we review several basic notions, mainly in order to establish our notation. For further details, the references should be consulted. In particular, [10] is our standard reference for automaton decompositions.

The symbol N denotes the set of all non-negative integers. An *alphabet* is a finite, non-empty set. Let X be an alphabet. Then X^* denotes the set of words over X including the empty word ε, and $X^+ = X^*\setminus\{\varepsilon\}$. For a word $w \in X^*$, $|w|$ denotes the length of w.

An *automaton* is a triple $\mathscr{A} = (A, X, \delta)$ with A a non-empty set, the *set of states*, X an alphabet, the *input alphabet*, and with $\delta : A \times X \to A$ the *transition function*. As usual, δ is extended to a function of $A \times X^*$ into A by requiring

$$\delta(a, w) = \begin{cases} a & \text{if } w = \varepsilon, \\ \delta(\delta(a, v), x) & \text{if } w = vx \text{ with } x \in X,\ v \in X^*. \end{cases}$$

Occasionally we also need to consider a *non-deterministic automaton*. In this case, the transitions are defined by a mapping δ of $A \times X$ into 2^A instead of into A.

Let $\mathscr{A} = (A, X, \delta)$ be an automaton. It is a *permutation automaton* if $\delta(a, x) \neq \delta(b, x)$ for all $a, b \in A$ with $a \neq b$ and for all $x \in X$. The automaton $\mathscr{A}$ is *commutative* if $\delta(a, xy) = \delta(a, yx)$ for all $a \in A$ and all $x, y \in X$. It is a *counter* if $|X| = 1$, say $X = \{x\}$, and $A = \{1, \ldots, n\}$ for some $n \in \mathrm{N}$, $n \geqslant 1$, such that

$$\delta(i, x) \equiv i + 1 \pmod{n}.$$

Finally, $\mathscr{A}$ is called a *reset automaton* if for some $x \in X$, one of the following two conditions obtains:
- $\delta(a, x) = a$ for all $a \in A$; or
- there is a state $a_x \in A$ such that $\delta(a, x) = a_x$ for all $a \in A$.

A reset automaton is *full* if, in addition to the above, it satisfies the following two conditions:
- there is an $x \in X$ with $\delta(a, x) = a$ for all $a \in A$; and
- for every $a \in A$ there is a symbol $x_a \in X$ such that $\delta(b, x_a) = a$ for all $b \in A$.

Reset automata and counters are standard building blocks for arbitrary automata.

For $k \in \mathrm{N}$, $k > 0$, and $i = 1, 2, \ldots, k$ let $\mathscr{A}_i = (A_i, X_i, \delta_i)$ be an automaton. Let X be an alphabet and φ be a mapping,

$$\varphi : A_1 \times \cdots \times A_k \times X \to X_1 \times \cdots X_k.$$

The (*general*) *product*

$$\mathscr{A}(A, X, \delta) = \prod_{i=1}^{k} \mathscr{A}_i[X, \varphi]$$

of the automata $\mathscr{A}_1, \ldots, \mathscr{A}_k$ is defined as follows:

$$A = A_1 \times \cdots \times A_k,$$

$$\delta((a_1, \ldots, a_k), x) = (\delta_1(a_1, x_1), \ldots, \delta(a_k, x_k))$$

where

$$(a_1, \ldots, a_k) \in A, \qquad x \in X,$$

and

$$(x_1, \ldots, x_k) = \varphi(a_1, \ldots, a_k, x).$$

The mapping φ is called the feed-back function of the product. To simplify notation we write

$$\varphi(\boldsymbol{a}, x) = (\varphi_1(\boldsymbol{a}, x), \ldots, \varphi_k(\boldsymbol{a}, x))$$

where $\boldsymbol{a} \in A$ and $x \in X$. The product is called an α_i-*product* if for $t = 1, \ldots, k$, each φ_t is independent of its jth component whenever $j \geq t + i$. Figure 2 illustrates the general product of automata.

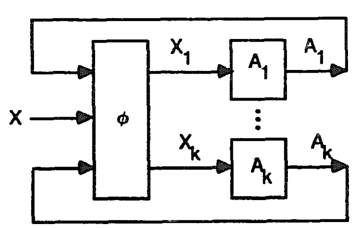

Fig. 2. The general product of automata.

In this paper we consider only the general product and α_i-products. Let β denote any of g (for "general") and α_i. If $\mathscr{K}$ is any class of automata, then $\mathrm{HSP}_\beta(\mathscr{K})$ denotes the closure of $\mathscr{K}$ under β-products, subautomata, and homomorphic images. Similarly, $\mathrm{ISP}_\beta(\mathscr{K})$ is the closure of $\mathscr{K}$ under β-products, subautomata, and isomorphisms. Let also $\mathscr{T}$ be a class of automata. The class $\mathscr{K}$ is said to be *homomorphically complete* for $\mathscr{T}$ with respect to β-products if $\mathscr{T} \subseteq \mathrm{HSP}_\beta(\mathscr{K})$. It is said to be homomorphically complete with respect to β-products if $\mathrm{HSP}_\beta(\mathscr{K})$ is the class of all automata. It is homomorphically complete if it is so with respect to the general product. Isomorphic completeness is defined in an analogous fashion using the operator ISP_β.

3. Soliton automata

Our formal model of soliton automata is based on graph theoretical notions. A *graph* is a pair $G=(N, E)$ with N a set, the set of *nodes*, and with $E \subseteq N \times N$ the set of *edges*. In this paper we consider finite, non-empty undirected graphs only; therefore, in the sequel we assume without special mention that N is finite and non-empty, and that $E^{-1} \subseteq E$, that is, for all nodes $n, n' \in N$ one has $(n', n) \in E$ if $(n, n') \in E$. However, if $(n, n') \in E$ then both (n, n') and (n', n) represent the same edge of G. Observe that with this definition any two nodes of a graph can be connected by at most one edge. A mapping $w: N \times N \to \mathbb{N}$ is called a *weight function* if

$$w(n, n') = \begin{cases} 0 & \text{for } (n, n') \notin E, \\ w(n', n) > 0 & \text{for } (n, n') \in E. \end{cases}$$

A triple $G=(N, E, w)$ with (N, E) a graph and w a weight function on (N, E) is called a *weighted graph*. Clearly, more general types of weight functions could be considered; however, for this paper the notion as introduced is general enough.

For a node $n \in N$ the set $V(n) = \{n' \mid (n, n') \in E\}$ is the *vicinity* (neighbourhood) of n, the integer $d(n) = |V(n)|$ is its *degree*, and

$$w(n) = \sum_{n' \in V(n)} w(n, n')$$

is its *weight*. A node n is said to be *isolated* if $d(n)=0$, *exterior* if $d(n)=1$, and *interior* if $d(n)>1$.

The following definition abstracts from the physico-chemical details of the examples of "soliton valves" built from polyacetylene chains as proposed in [1]. A more general definition is conceivable, but would almost certainly lead too far away from chemical and physical facts as currently available.

Definition 3.1. A *soliton graph* is a weighted graph $G=(N, E, w)$ which satisfies the following conditions:

(a) G has no loops; that is $(n, n) \notin E$ for all $n \in N$;

(b) every component (that is, maximal connected subgraph) of G has at least one exterior node;

(c) for every $n \in N$ one has $1 \leq d(n) \leq 3$;

(d) if n is an exterior node then $w(n) \in \{1, 2\}$;

(e) for every $n \in N$ with $d(n) \in \{2, 3\}$ one has $w(n) = d(n) + 1$.

A soliton graph $G=(N, E, w)$ models the "soliton valves" of [1] as follows. Each interior node n represents a C atom or a C–H group depending on whether $d(n)$ is 3 or 2, respectively. An edge (n, n') of weight $i \in \{1, 2\}$ represents a (CH)-chain with alternating double and single bonds which connects the C atoms of n and n'

and which begins and ends with an i-fold bond. As the length of such chains does not affect the logic of the model we usually draw them as length 1 chains; physico-chemical reasons may require different lengths for actual realizations. Finally, exterior nodes represent the connection to surrounding structures. Certain proposals as to their chemical realization can be found in [1]. Figure 3 shows an example of a soliton graph and a possible chemical interpretation. The weights of edges are indicated by single and double lines.

A simple case of a "conjugated system" in a soliton graph would be a path $n_0, n_1, \ldots, n_k$ such that $|w(n_i, n_{i+1}) - w(n_{i+1}, n_{i+2})| = 1$ for $i = 0, 1, \ldots, k-2$. That is, such a path would have alternating single and double bonds in the chemical interpretation. A soliton propagating along such a path will exchange single and double bonds. In our model this corresponds to exchanging weights 1 and 2.

However, this is too simple for a correct definition of a model of soliton propagation. Figure 4 shows an example adopted from [1] in which edges would be traversed more than once on a path. Thus using statically conjugated systems as the basis for a definition of the switching behaviour that models soliton movement, would be inadequate. The following more complicated definition is required.

Definition 3.2. Let $G = (N, E, w)$ be a soliton graph. A path $n_0, n_1, \ldots, n_k$ of G is called a *partial soliton path* if the following conditions hold:

(a) n_0 is an exterior node;

(b) $n_1, n_2, \ldots, n_{k-1}$ are interior nodes;

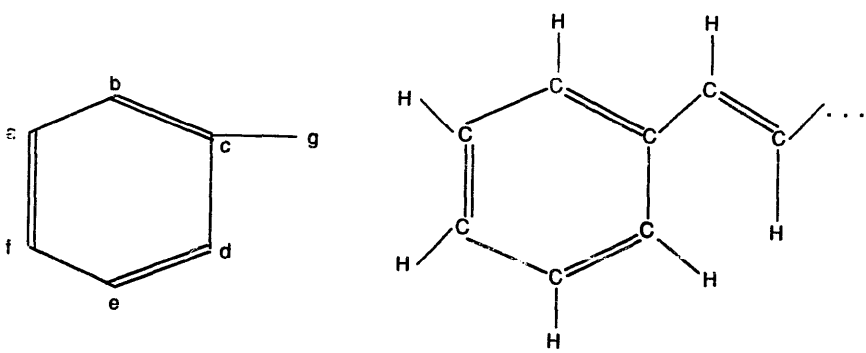

Fig. 3. A soliton graph with one of its interpretations.

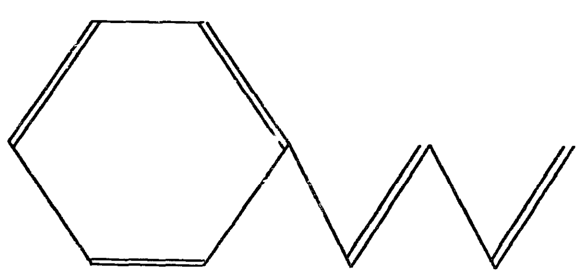

Fig. 4. A soliton graph with an edge traversed several times [1].

(c) there is a sequence $G_0, G_1, \ldots, G_k$ of weighted graphs $G_i = (N, F, w_i)$ which can be constructed as follows:

(c1) $G_0 = G$;

(c2) for $i = 0, 1, \ldots, k-2$ the graph $G_{i+1} = (N, E, w_{i+1})$ is defined if and only if G_i is defined and $|w_i(n_i, n_{i+1}) - w_i(n_{i+1}, n_{i+2})| = 1$. In this case,

$$w_{i+1}(n, n') = \begin{cases} w_i(n, n') & \text{if } (n, n') \neq (n_i, n_{i+1}), \\ 3 - w_i(n_i, n_{i+1}) & \text{if } (n, n') = (n_i, n_{i+1}) \end{cases}$$

for all $n, n' \in N$.

(c3) G_k is defined if and only if G_{k-1} is defined. In this case,

$$w_k(n, n') = \begin{cases} w_{k-1}(n, n') & \text{if } (n, n') \neq (n_{k-1}, n_k), \\ 3 - w_{k-1}(n_{k-1}, n_k) & \text{if } (n, n') = (n_{k-1}, n_k) \end{cases}$$

for all $n, n' \in N$.

Such a partial soliton path is called a (*total*) *soliton path* if n_k is an exterior node.

The example in Fig. 5 illustrates the definition of a soliton path. G_0 is the initial graph. We then consider the path $2xyzxy3$ resulting in the sequence $G_1, \ldots, G_6$ of graphs. In each of them the "position of the soliton" is indicated by an arrow. Note that the intermediate graphs $G_1, \ldots, G_5$ are not necessarily soliton graphs. For the interpretation, again one has to keep in mind that single edges in the graph may well represent more complex structures—like long chains, for instance; this would have an effect on the timing of such a system.

Given a soliton graph $G = (N, E, w)$ and a pair of exterior nodes $n, n' \in N$, let $S(G, n, n')$ be the set of weighted graphs G' which can be obtained as $G' = G_k$ for some soliton path $n = n_0, \ldots, n_k = n'$. We say that G' *is generated by a transition from G* if $G' \in S(G, n, n')$ for some exterior nodes $n, n' \in N$. The following lemmas state that our definitions so far make sense.

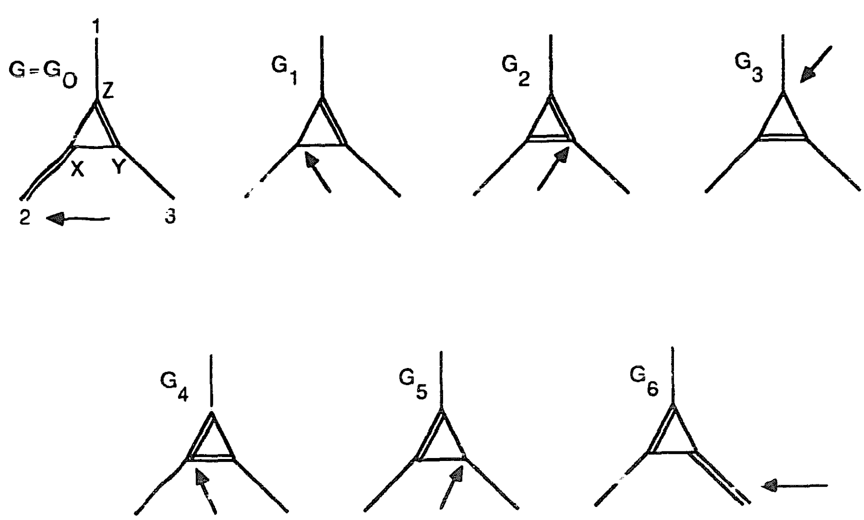

Fig. 5. Example of a soliton path and the corresponding sequence of graphs.

Lemma 3.3 (Dassow and Jürgensen [3]). *Let G be a soliton graph, and let $G' \in S(G, n, n')$ for some exterior nodes of G. Then G' is also a soliton graph.*

Lemma 3.4 (Dassow and Jürgensen [3]). *Let G be a soliton graph. If G' is obtained from G by a transition, then G is also obtained from G' by a transition.*

For a set of soliton graphs $\mathscr{H}$ consider the sequence $\mathscr{H}_0 = \mathscr{H}, \mathscr{H}_1, \mathscr{H}_2, \ldots$ where for $i = 0, 1, 2, \ldots$ the set $\mathscr{H}_{i+1}$ is the union of $\mathscr{H}_i$ with the set of those soliton graphs that can be obtained by a transition from a graph in $\mathscr{H}_i$. The particular case of interest in this paper is that of $\mathscr{H} = \{G\}$. In this case let

$$S(G) = \bigcup_{i=0}^{\infty} \mathscr{H}_i.$$

Obviously, as G is finite, $S(G)$ is also finite and, in fact, can be obtained in finitely many computational steps.

Lemma 3.5 (Dassow and Jürgensen [3]). *Let G be a soliton graph, and let $G' \in S(G)$. Then $S(G') = S(G)$.*

Definition 3.6. Let G be a soliton graph with X its set of exterior nodes. The *soliton automaton* based on G is defined as the non-deterministic automaton

$$\mathscr{A}(G) = (S(G), X \times X, \delta)$$

subject to the following conditions:

(a) $S(G)$ is the set of states;

(b) $X \times X$ is the input alphabet;

(c) $\delta : S(G) \times X \times X \to 2^{S(G)}$ is the transition function with

$$\delta(G', n, n') = \begin{cases} S(G', n, n') & \text{if } S(G', n, n') \neq \emptyset, \\ \{G'\} & \text{otherwise} \end{cases}$$

for $G' \in S(G)$ and $n, n' \in X$.

Usually, a soliton automaton will have several equivalent input symbols, that is, input symbols which cause exactly the same state transitions. For instance, the symbols (n, n') and (n', n) for $n, n' \in N$ are always equivalent. In the sequel, such equivalent inputs will not be mentioned explicitly.

Note that the empty path is never considered a soliton path. Hence, if n is an exterior node the set $S(G, n, n)$ resulting from soliton paths starting and ending at n will be non-empty only if there are non-empty cyclic soliton paths from n to itself. Otherwise, the transition caused by (n, n) is defined as the identity transition.

Example 3.7. Consider the graph G in Fig. 6(a). One obtains the transitions shown in Fig. 6(b). The resulting automaton $\mathcal{A}$ has the following transition function:

	a	b	c
$(1,1)$	a	c	b
$(1,2)$	b, c	a	a

The automaton $\mathcal{A}$ is non-deterministic in the usual sense of the term. However, it also exhibits a slightly different kind of non-determinism. For the same input symbol, different paths can be used which, nevertheless, result in the same state transition. This distinction is made precise in the following definition.

Definition 3.8. Let G be a soliton graph. G is called *deterministic* if $|S(G', n, n')| \leq 1$ for all $G' \in S(G)$ and all exterior nodes $n, n' \in N$. It is called *strongly deterministic* if for every $G' \in S(G)$ and for every pair of exterior nodes $n, n' \in N$ there is at most one soliton path from n to n' in G'.

Observe that the soliton automaton $\mathcal{A}(G)$ based on a soliton graph G is deterministic in the usual automaton theoretic sense if and only if G is deterministic. By slight abuse of definitions, the soliton automaton $\mathcal{A}(G)$ is called *strongly deterministic* if G is strongly deterministic.

Obviously the connected components of a soliton graph act as "independent units" in the corresponding soliton automaton. However, connectedness is insufficient as a property to guarantee that the resulting automaton be without such "independent subunits". Indeed, such "independent subunits" exist whenever a connected soliton graph G contains subgraphs G_1 and G_2 which are not joined by any soliton path.

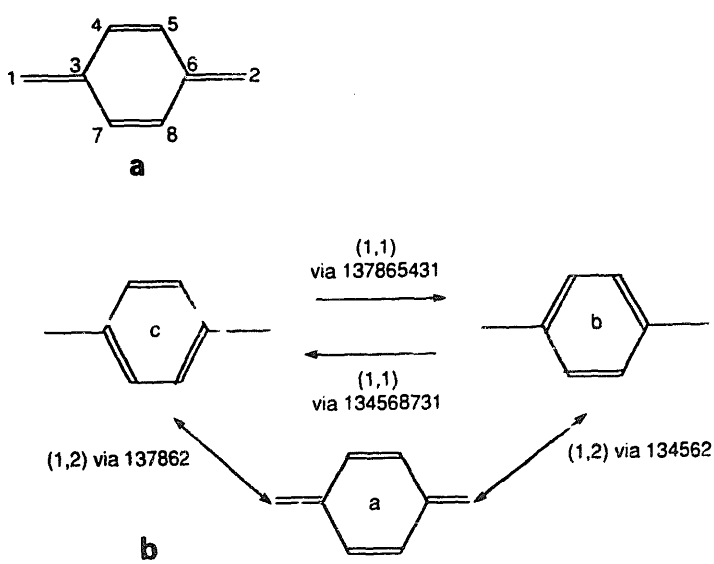

Fig. 6.

Definition 3.9. Let $G = (N, E, w)$ be a soliton graph. An edge $(n, n') \in E$ is said to be *impervious* if it is not contained in any partial soliton path of G. A path of G is called impervious if each of its edges is impervious.

In Fig. 7(a) we show two examples of soliton graphs; in either the path 1234 is impervious. An impervious path can always be extended, if necessary, to end and begin with nodes of degree 3. A path $n_0, n_1, \ldots, n_k$ is a *basic impervious path* if $d(n_0) = d(n_k) = 3$ and $d(n_1) = \cdots = d(n_{k-1}) = 2$. The next lemma shows that basic impervious paths can be omitted from a soliton graph without affecting its behaviour as an automaton.

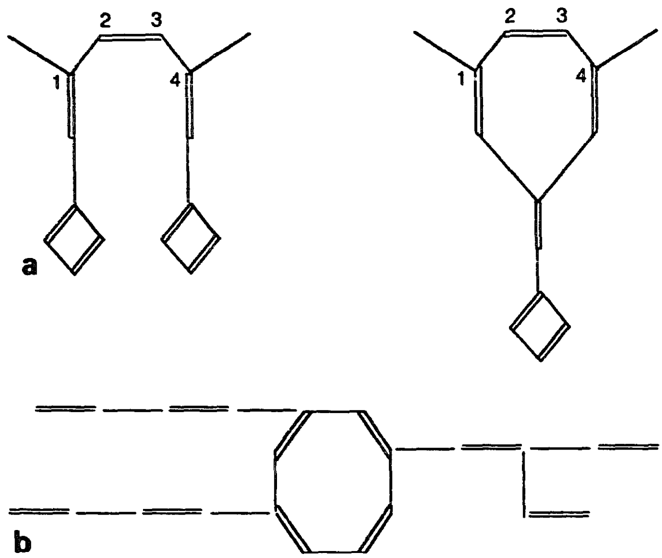

Fig. 7. (a) Soliton graphs with impervious paths. (b) A chestnut.

Lemma 3.10. *Let $G = (N, E, w)$ be a deterministic soliton graph containing a basic impervious path $n_0, \ldots, n_k$. Let*

$$N' = N \setminus \{n_1, \ldots, n_{k-1}\},$$

$$E' = E \setminus \{(n_0, n_1), (n_1, n_2), \ldots, (n_{k-1}, n_k)\},$$

and let w' be the restriction of w to E'. Then $G' = (N', E', w')$ is a soliton graph satisfying $\mathcal{A}(G) \simeq \mathcal{A}(G')$.

Proof. The statement results from the proof in [3, Lemma 4.4]. □

Let G be a soliton graph. Using the result of the lemma on its connected components iteratively, one obtains a *reduced soliton graph* H with the same set X of exterior nodes such that the connected components $H_1, \ldots, H_r$ of H themselves contain no basic impervious paths. Indeed, H is uniquely determined by G. The decomposition $H_1, \ldots, H_r$ of H is called the *soliton decomposition* of G. A soliton graph is called *indecomposable* if it is connected and contains no impervious edges.

An indecomposable soliton graph which consists of a single cycle of even length and some paths leading into it, as shown, is called a *chestnut* (Fig. 7(b)). The only condition on the way in which the paths enter the cycle is the following. Entry points of different paths entering the cycle have an even distance; paths leading to the cycle may meet only at even distances from entry into the cycle. We quote a characterization of strongly deterministic indecomposable soliton graphs and a description of the transition monoids of strongly deterministic automata from [3].

Theorem 3.11. *Let $G=(N, E, w)$ be an indecomposable soliton graph. Then G is strongly deterministic if and only if G is a chestnut or (N, E) is a tree.*

Theorem 3.12. *The transition monoid of a strongly deterministic soliton automaton is a direct product of primitive permutation groups which are generated by involutorial elements.*

In the sequel, let $\mathcal{S}$ denote the class of all strongly deterministic soliton automata. The following subclass of $\mathcal{S}$ is of particular importance. For $n \in \mathrm{N}$, $n \geq 1$, let $T_n = (N_n, E_n, w_n)$ with

$$N_n = \{1, \ldots, 2n+1\} \cup \{1', \ldots, n'\},$$

$$E_n = \{(i, i+1) \mid i = 1, \ldots, 2n\} \cup \{(2i, i') \mid i = 1, \ldots, n\},$$

and

$$w_n(i, j) = \begin{cases} 2 & \text{if } j = i+1 \text{ and } i \text{ is odd}, \\ 1 & \text{if } j = i+1 \text{ and } i \text{ is even}, \\ 1 & \text{if } i = 2k \text{ and } j = k', \\ 0 & \text{otherwise}. \end{cases}$$

The soliton tree T_n is illustrated in Fig. 8. Let $\mathcal{A}_n = \mathcal{A}(T_n)$, and let $\mathcal{G} = \{\mathcal{A}_n \mid n = 1, 2, \ldots\}$. For $i = 1, \ldots, n$ let $T_{n,i}$ denote the tree obtained from T_n by input $(1, i')$. It is known that the transition monoid of $\mathcal{A}_n$ is the symmetric group $\mathcal{S}_n$.

The following rather technical point requires special mention: very often we utilize the existence of an input of the form (i, i) to a soliton automaton to induce the identity permutation. If this is unacceptable from an application point of view, the arguments would have to be modified slightly. However, in essence, all proofs remain valid if this particular kind of input is unavailable.

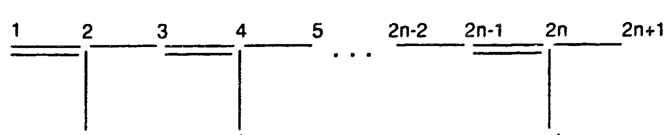

Fig. 8. Soliton tree T_n.

4. Homomorphic representation by α_0-products

It is an immediate consequence of the definition that every strongly deterministic soliton automaton is a permutation automaton [3]. The class of permutation automata is closed under α_0-products, homomorphisms, and the operation of taking subautomata. Therefore, the class $\mathrm{HSP}_{\alpha_0}(\mathcal{S})$ contains permutation automata only. This leads to the following observation.

Remark 4.1. The class $\mathcal{S}$ is not homomorphically complete with respect to the α_0-product.

Given this fact, it is natural to ask the question which class of automata can actually be represented homomorphically by α_0-products of strongly deterministic soliton automata. We prove in this section, that $\mathcal{S}$ is homomorphically complete for the class of all commutative permutation automata with respect to the α_0-product. In fact, a slightly stronger statement is shown to hold: already the class $\mathcal{G} \subseteq \mathcal{S}$ is homomorphically complete for the class of commutative permutation automata with respect to the α_0-product.

For every $n \in \mathrm{N}$ let $\mathcal{C}_n = (C_n, X, \gamma_n)$ denote the automaton defined by

$$C_n = \{1, \ldots, n\}, \qquad X = \{x, y\},$$

$$\gamma_n(i, x) = \begin{cases} i+1 & \text{if } i < n, \\ i & \text{if } i = n, \end{cases}$$

$$\gamma_n(i, y) = i$$

for $i \in C_n$. The following result is an immediate consequence of [12, Theorem 2].

Lemma 4.2. *The class $\{\mathcal{C}_p \mid p$ is a prime$\}$ is homomorphically complete for the class of all commutative permutation automata with respect to the α_0-product.*

We can now proceed to state the main result of this section.

Theorem 4.3. *For every $p \in \mathrm{N}$, $p \geq 1$, one has $\mathcal{C}_p \in \mathrm{HSP}_{\alpha_0}(\mathcal{G})$.*

Proof. For $p = 1$ there is nothing to prove. Consider the case of $p = 2$. For any n, the soliton tree T_n allows for the following transitions in particular:

$$\delta_n(T_n, (1, 1')) = T_{n,1}, \qquad \delta_n(T_{n,1}, (1, 1')) = T_n,$$

$$\delta_n(T_n, (1, 1)) = T_n, \qquad \delta_n(T_{n,1}, (1, 1)) = T_{n,1}.$$

Therefore, any single-factor α_0-product of $\mathcal{A}_n$ represents $\mathcal{C}_2$ isomorphically.

Next, we show that also every $\mathcal{C}_p$ with $p = 2^m$ for some integer m, $n > 1$, can be represented homomorphically by an α_0-power of $\mathcal{C}_2$ and, thus, by an α_0-power of

$\mathscr{A}_n$. We use the construction given in the proof in [9, Proposition 6]. Assume that $\mathscr{C}_l \in \mathrm{HSP}_{\alpha_0}(\mathscr{G})$ with $l = p/2$. Consider the α_0-product

$$\mathscr{A} = (A, X, \delta) = (\mathscr{C}_l \times \mathscr{C}_2)[X, \varphi]$$

with

$$X = \{x, y\}, \qquad \varphi_1(x) = x, \qquad \varphi_1(y) = y,$$

$$\varphi_2(i, x) = \begin{cases} x & \text{if } i = 1, \\ y & \text{if } 1 < i \leqslant l, \end{cases} \qquad \varphi_2(i, y) = y$$

for $i = 1, 2, \ldots, l$. Then $\mathscr{A}$ is isomorphic with $\mathscr{C}_p$.

Finally, consider the case of arbitrary p, $p > 2$. Choose arbitrary integers n, m with $n \geqslant p$ and $2^m > p$, and take the input word

$$v = (1, 1')(1, 2') \ldots (1, p')(1, 1')$$

of the strongly deterministic soliton automaton $\mathscr{A}_n$. Clearly,

$$\delta_n(T_{n,i}, v) = \begin{cases} T_{n,i+1} & \text{if } i < p, \\ T_{n,1} & \text{if } i = p, \end{cases} \qquad \delta_n(T_{n,i}, (1, 1)) = T_{n,i}$$

for $i = 1, \ldots, p$. Now consider the α_0-product

$$\mathscr{A} = (A, X, \delta) = (\mathscr{C}_l \times \mathscr{A}_n)[X, \varphi]$$

where

$$l = 2^m, \qquad X = \{x, y\}, \qquad \varphi_1(x) = x, \qquad \varphi_1(y) = y,$$

$$\varphi_2(i, x) = \begin{cases} (1, i') & \text{if } 1 \leqslant i \leqslant p, \\ (1, 1') & \text{if } i = p + 1, \\ (1, 1) & \text{if } p + 1 < i \leqslant l, \end{cases} \qquad \varphi_2(i, y) = (1, 1).$$

It follows that the set

$$\{(j, T_{n,i}) \mid 1 \leqslant j \leqslant l, 1 \leqslant i \leqslant p\}$$

defines a subautomaton of $\mathscr{A}$ which is isomorphic with $\mathscr{C}_t$ where $t = lp$. Finally, $\mathscr{C}_p$ is a homomorphic image of $\mathscr{C}_t$. $\square$

As an immediate consequence of these results one shows that the class of strongly deterministic soliton automata is homomorphically complete for the class of commutative permutation automata with respect to the α_0-product. In fact, we formulate a slightly stronger version of this statement.

Corollary 4.4. *The class $\mathscr{G}$ is homomorphically complete for the class of commutative permutation automata with respect to the α_0-product.*

It is an open problem, whether the statement of Corollary 4.4 holds true without the commutativity assumption. This condition enters the present version of the result via Lemma 4.2.

5. Homomorphic representations by α_1-products

The class of strongly deterministic soliton automata turns out to be homomorphically complete with respect to α_1-products. This is in contrast to the situation with α_0-products. In fact, only a special kind of α_1-products is required. These assertions are proved in the present section.

The following is a well-known classical result of automaton theory [13].

Theorem 5.1. *Every automaton can be represented homomorphically by an α_0-product of permutation automata and two-state reset automata.*

Let X and Y be alphabets, let n be an integer, $n>0$, and let $\tau: X^n \to Y^n$ be a mapping. Define the automaton $\mathcal{R}_\tau = (R_\tau, X, \delta_\tau)$ by

$$R_\tau = \{(p, q) \mid (p, q) \in X^* \times Y^*, 1 \leq |p|, |q| \leq n, |p| + |q| = n+1\}$$

and

$$\delta_\tau((p, yq), x) = \begin{cases} (px, q) & \text{if } |p| < n, \\ (x, \tau(p)) & \text{if } |p| = n, \end{cases}$$

for $(p, yq) \in R_\tau$, $y \in Y$, and $x \in X$. The following result is taken from [7] (see also [10]).

Lemma 5.2. *For every $n>0$ and $\tau: X^n \to Y^n$, the automaton $\mathcal{R}_\tau$ can be represented homomorphically by an α_0-product of an n-state counter and two-state reset automata.*

Lemma 5.3 (Gécseg [10]). *Let $\mathcal{A} = (A, X, \delta_A)$ and $\mathcal{B} = (B, X, \delta_B)$ be automata for which there exist an integer n, $n>0$, and a mapping $\tau: X^n \to Y^n$ satisfying the following two conditions:*

(i) $\delta_A(a, p) = \delta_B(a, \tau(p))$ *for all* $a \in A$ *and* $p \in X^n$;

(ii) $\{\delta_A(\delta_B(a, q), p) \mid a \in A, (p, q) \in R_\tau\} = A$.

Then there is an α_0-product of $\mathcal{R}_\tau$ by $\mathcal{B}$ which represents $\mathcal{A}$ homomorphically.

Note that the decompositions described by Theorem 5.1 and Lemmas 5.2 and 5.3 can be obtained effectively. In view of these results, we can deal with reset automata and permutation automata, separately.

Lemma 5.4. *Any reset automaton $\mathcal{A} = (\{a_1, a_2\}, X, \delta_A)$ can be represented homomorphically by a single-factor α_1-product of any strongly deterministic soliton automaton $\mathcal{A}_n \in \mathcal{G}$ with $n \geq 1$.*

Proof. Take any $\mathcal{A}_n \in \mathcal{G}$ with $n \geq 1$ and form the α_1-product

$$\mathcal{B} = (B, X, \delta_B) = \mathcal{A}_n[X, \varphi]$$

with φ defined as follows for $x \in X$:

$$\varphi(T_n, x) = \begin{cases} (1, 1') & \text{if } \delta_A(a_1, x) = a_2, \\ (1, 1) & \text{if } \delta_A(a_1, x) = a_1, \end{cases}$$

$$\varphi(T_{n,1}, x) = \begin{cases} (1, 1') & \text{if } \delta_A(a_2, x) = a_1, \\ (1, 1) & \text{if } \delta_A(a_2, x) = a_2. \end{cases}$$

In all other cases, φ is defined in an arbitrary way.

Now consider the mapping

$$\alpha : \{a_1, a_2\} \to S(T_n) : \begin{cases} a_1 \mapsto T_n, \\ a_2 \mapsto T_{n,1}. \end{cases}$$

One easily verifies that α is an isomorphism of $\mathscr{A}$ into $\mathscr{B}$. □

We can now prove the main result of this section.

Theorem 5.5. *The class $\mathscr{G}$ is homomorphically complete with respect to the α_1-product.*

Proof. Because of Theorem 5.1 and Lemma 5.4, it is sufficient to prove that every permutation automaton is in $\mathrm{HSP}_{\alpha_1}(\mathscr{G})$.

By Theorem 4.3, every n-state counter is in $\mathrm{HSP}_{\alpha_0}(\mathscr{G})$. Let $\mathscr{A} = (A, X, \delta_A)$ be an arbitrary permutation automaton. We may assume that

$$A = \{T_{m,1}, \ldots, T_{m,m}\}$$

for some $m \in \mathbb{N}$. If $(T_{m,i_1} \cdots T_{m,i_l})$ is a cycle of elements of A, then the word

$$(1, i_1') \ldots (1, i_l')(1, i_1')$$

induces this cycle in $\mathscr{A}_m$. In fact, if this word is applied to T_{m,i_j} then $(1, i_1') \ldots (1, i_{j-1}')$ acts as the identity mapping. The input symbol $(1, i_j')$ then results in T_m, $(1, i_{j+1}')$ yields $T_{m,i_{j+1}}$, and $(1, i_{j+2}') \ldots (1, i_l')(1, i_1')$ does not change this state. Moreover, the input $(1, 1)$ induces the identity permutation of A. Using the representation of permutations of A by disjoint cycles, every permutation of A can be induced by a word over the alphabet $Y = Y' \cup \{(1, 1)\}$ where

$$Y' = \{(1, 1'), \ldots, (1, m')\}.$$

Furthermore, every such word can be assumed to be of length n where

$$n = \begin{cases} 3k & \text{if } m = 2k, \\ 3k+1 & \text{if } m = 2k+1. \end{cases}$$

Now define $\tau : X^n \to Y^n$ in such a way that the permutations induced by $p \in X^n$ in $\mathscr{A}$ and by $\tau(p)$ in $\mathscr{A}_m$ are the same. Obviously, such a τ exists. Letting $\mathscr{B} = \mathscr{A}_m$ in Lemma 5.3, τ satisfies conditions (i) and (ii) of that lemma. Thus, the automaton

$\mathscr{A}$ can be represented homomorphically by an α_0-product of $\mathscr{R}_\tau$ by $\mathscr{A}_m$. By Lemma 5.2, this proves that $\mathscr{A} \in \mathrm{HSP}_{\alpha_1}(\mathscr{G})$. □

6. Homomorphic representation by α_i-products with $i \geqslant 2$

In [8], it is shown that the α_2-product is homomorphically equivalent to the general product. Therefore, in studying homomorphic representations by α_i-products of strongly deterministic soliton automata with $i \geqslant 2$, we need to consider general products only. We quote two results from [14].

Lemma 6.1. *A class K of automata is homomorphically complete with respect to the general product if and only if K contains an automaton* $\mathscr{A} = (A, X, \delta)$ *with*

$$\delta(a, x_1) \neq \delta(a, x_2) \quad \textit{and} \quad \delta(\delta(a, x_1), p_1) = \delta(\delta(a, x_2), p_2) = a$$

for some $a \in A$, $x_1, x_2 \in X$, *and* $p_1, p_2 \in X^*$.

Lemma 6.2. *A class K of automata is isomorphically complete with respect to the general product if and only if K contains an automaton* $\mathscr{A} = (A, X, \delta)$ *with*

$$\delta(a_1, x_1) = \delta(a_2, x_3) = a_2 \neq \delta(a_2, x_2) = \delta(a_1, x_4) = a_1$$

for some $a_1, a_2 \in A$ *and some* $x_1, x_2, x_3, x_4 \in X$.

Now we give a necessary and sufficient condition for a class of strongly deterministic soliton automata to be homomorphically complete with respect to the general product.

Theorem 6.3. *A class K of strongly deterministic soliton automata is homomorphically complete with respect to the general product (or the* α_i*-product with* $i \geqslant 2$*) if and only if K contains an automaton whose underlying soliton graph G satisfies one of the following three conditions:*

(i) *the soliton decomposition of G consists of at least two components;*

(ii) *the soliton decomposition of G consists of a single tree with at least two nodes;*

(iii) *the soliton decomposition of G consists of a single chestnut with at least two exterior nodes.*

Proof. Suppose that no soliton graph G with $\mathscr{A}(G) \in K$ satisfies any of the conditions. Thus, if G is a soliton graph with $\mathscr{A}(G) \in K$ then the soliton decomposition of G has a single component which is a chestnut with a single exterior node. The automaton $\mathscr{A}(G)$ has two states and a single input letter, $(1, 1)$ say, which induces the cyclic permutation on the state set. By Lemma 6.1, the class K is not homomorphically complete. This proves that the conditions are necessary.

Now consider the converse. Consider $\mathscr{A}(G) \in K$ with $\mathscr{A}(G) = (S(G), X, \delta)$.

First suppose that G satisfies (i). In this case there are exterior nodes n_1, n_2, n_3 of G such that n_1 and n_2 are connected by a soliton path whereas n_3 is in a component of the soliton decomposition of G which does not contain n_1, n_2. Let $a_1 = G$, $a_2 = \delta(G, (n_1, n_2))$, $x_1 = (n_1, n_2) = x_2$, and $x_3 = (n_1, n_3) = x_4$. Clearly, $a_1 \neq a_2$, and the condition of Lemma 6.2 is satisfied. Therefore K is isomorphically complete.

If G satisfies (ii), G has at least two distinct exterior nodes n_1 and n_2 which are connected by a soliton path. Let $a_1 = G$, $a_2 = \delta(G, (n_1, n_2))$, $x_1 = (n_1, n_2) = x_2$, and $x_3 = (n_1, n_1) = x_4$. Again, by the lemma, K is isomorphically complete.

Finally, if G satisfies (iii), G has at least two distinct exterior nodes n_1 and n_2. Furthermore, there is a soliton path joining n_1 with itself, but no soliton path joining n_1 with n_2. Thus, with the roles of (n_1, n_2) and (n_1, n_1) exchanged, the arguments for case (ii) apply; that is, K is isomorphically complete. □

Suppose that K is a class of strongly deterministic soliton automata which is homomorphically complete with respect to the general product. Then there is a soliton graph G with $\mathscr{A}(G) \in K$ which satisfies one of the conditions of Theorem 6.3. The proof of this theorem shows that in such a case K is isomorphically complete. This proves the following statement.

Corollary 6.4. *A class of strongly deterministic soliton automata is homomorphically complete with respect to the general product if and only if it is isomorphically complete with respect to the general product.*

7. Isomorphic representation by α_i-products

As far as homomorphic and isomorphic completeness of classes of strongly deterministic soliton automata is concerned, the only case which still has to be considered is that of isomorphic completeness with respect to the α_i-product. In this section we prove that there is no class of isomorphically complete strongly deterministic soliton automata with respect to any α_i-product. The following is a slight modification of a result of [15].

Lemma 7.1. *A class K of automata is isomorphically complete with respect to the α_i-product with $i \geq 1$ if and only if for every $n \in \mathrm{N}$, $n \geq 1$, there is an automaton $\mathscr{A} = (A, X, \delta) \in K$ which satisfies the following condition: there are n distinct states $a_1, \ldots, a_n \in A$ such that for every i and j with $1 \leq i, j \leq n$ there is a symbol $x \in X$ with $\delta(a_i, x) = a_j$.*

Theorem 7.2. *Let $i \in \mathrm{N}$. There is no class of strongly deterministic soliton automata which is isomorphically complete with respect to the α_i-product.*

Proof. Suppose that K is a class of strongly deterministic soliton automata which is isomorphically complete with respect to the α_i-product. Without loss of generality, we may assume that $i \geq 1$. Let $\mathscr{A}(G) = (S(G), X \times X, \delta)$ be an automaton in K

which satisfies the condition of the lemma for $n=3$. We may also assume that G is reduced. As a chestnut has only two states, G is either a tree or consists of at least two components.

Now let $G_1, G_2, G_3 \in S(G)$ be distinct states of $\mathscr{A}(G)$ satisfying the condition of the lemma. Thus, there are exterior nodes n_1, n_2, m_1, m_2 such that

$$\delta(G_1, (n_1, n_2)) = G_2 \quad \text{and} \quad \delta(G_1, (m_1, m_2)) = G_3.$$

Consequently, there are soliton paths joining n_1 with n_2 and m_1 with m_2 in G_1.

If n_1 and m_1 are in different components of G then there is no input (i, j) with $\delta(G_2, (i, j)) = G_3$. Therefore, G is a single tree. But then it is obvious that there is no direct transition from G_2 to G_3, a contradiction! □

8. Concluding remarks

In [3, 4] it was shown that strongly deterministic soliton automata give rise to a rather restricted class of transition semigroups only. This can be viewed as a first, though crude, description of the computational power of strongly deterministic soliton automata. In the preceding chapters, an analysis based on a more realistic notion of automaton simulation has determined the computational power of strongly deterministic soliton automata more accurately.

As far as homomorphic representation is concerned, loop-free products are insufficient to simulate all automata; on the other hand, they suffice to simulate all commutative permutation automata. It is an open question, whether this result can be extended to include all permutation automata. On the other hand, already the availability of self-loops allows one to represent any automaton by a product of strongly deterministic soliton automata.

This rather favourable situation is in direct contrast to that of isomorphic representation. For the general product, that is, the product allowing for arbitrary feed-backs, the notions of homomorphic and isomorphic completeness coincide. However, for products with bounded length of feed-back, there are no isomorphically complete classes of strongly deterministic soliton automata.

The practical interpretation of these findings is ambiguous: one could say that the product representations are far too large to be feasible; one could also argue that given the miniature size of soliton automata, the product size does not matter at all. An even more detailed representation theory would be necessary to address the complexity questions.

Note added in proof

As stated, Lemma 3.10 is not correct. This does not affect the rest of the paper. A corrected formulation is contained in [5].

References

[1] F.L. Carter, Conformational switching at the molecular level, in: F.L. Carter, ed., *Molecular Electronic Devices* (Marcel Dekker, New York, 1982) 51–71.

[2] F.L. Carter (ed.), *Molecular Electronic Devices* (Marcel Dekker, New York, 1982).

[3] J. Dassow and H. Jürgensen, Soliton automata, *J. Comput. System Sci.*, to appear.

[4] J. Dassow and H. Jürgensen, Soliton automata, in: L. Budach, R.G. Bukharajev and O.B. Lupanov, eds., *Fundamentals of Computing Theory, Proc. Internat. Conf. FCT '87*, Kazan, USSR June 22–26, 1987, Lecture Notes in Computer Science **278** (Springer, Berlin, 1987) 95–102.

[5] J. Dassow and H. Jürgensen, *Deterministic Soliton Automata*, in preparation.

[6] A.S. Davydov, *Solitons in Molecular Systems* (D. Reidel, Dordrecht, 1985).

[7] P. Dömösi, On cascade products of standard automata, in: *Automata, Languages, and Systems: Conference Proceedings*, Salgótarján, Department of Mathematics, Karl Marx University of Economics, Budapest, DM 84-2 (1984) 37–45.

[8] Z. Ésik, Homomorphically complete classes of automata with respect to the α_2-product, *Acta Sci. Math.* **48** (1985) 135–141.

[9] F. Gécseg and Z. Ésik, On a representation of tree automata, *Fund. Inform.*, to appear.

[10] F. Gécseg, *Products of Automata*, EATCS Monographs on Theoretical Computer Science 7 (Springer, Berlin, 1986).

[11] F. Gécseg and I. Peák, *Algebraic Theory of Automata* (Akadémiai Kiadó, Budapest, 1972).

[12] F. Gécseg, Composition of automata, in: *Automata, Languages, and Programming, 2nd Colloquium*, Saarbrücken, 1974, Lecture Notes in Computer Science **14** (Springer, Berlin, 1974) 351–363.

[13] A. Ginzburg, *Algebraic Theory of Automata* (Academic Press, London, 1968).

[14] V.M. Gluškov, Abstract theory of automata (in Russian), *Uspekhi Mat. Nauk* **16:5**(101) (1961) 3–62; correction, *ibid.* **17:2**(104) (1962) 270.

[15] B. Imreh, On α_i-products of automata, *Acta Cybernet.* **3** (1978) 301–307.